AF/2186
IFW

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Application No.	09/122,349
Filing Date	July 24, 1998
First Named Inventor	Lance Hacking
Art Unit	2186
Examiner Name	Tran, D.
Attorney Docket Number	42390P5965
Total Number of Pages in This Submission	26

ENCLOSURES *(check all that apply)*

☒ Fee Transmittal Form
- ☒ Fee Attached

☐ Amendment / Response
- ☐ After Final
- ☐ Affidavits/declaration(s)

☐ Extension of Time Request

☐ Express Abandonment Request

☐ Information Disclosure Statement
- ☐ PTO/SB/08

☐ Certified Copy of Priority Document(s)

☐ Response to Missing Parts/ Incomplete Application
- ☐ Basic Filing Fee
- ☐ Declaration/POA
- ☐ Response to Missing Parts under 37 CFR 1.52 or 1.53

☐ Drawing(s)

☐ Licensing-related Papers

☐ Petition

☐ Petition to Convert a Provisional Application

☐ Power of Attorney, Revocation Change of Correspondence Address

☐ Terminal Disclaimer

☐ Request for Refund

☐ CD, Number of CD(s)

☐ After Allowance Communication to Group

☐ Appeal Communication to Board of Appeals and Interferences

☒ Appeal Communication to Group **(Appeal Notice, Brief, Reply Brief)**

☐ Proprietary Information

☐ Status Letter

☒ Other Enclosure(s) *(please identify below)*:

Return receipt postcard

Remarks

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm or Individual name	Joseph Lutz, Reg. No. 43,765 BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP
Signature	
Date	November 8, 2004

CERTIFICATE OF MAILING/TRANSMISSION

I hereby certify that this correspondence is being deposited with the United States Postal Service on the date shown below with sufficient postage as first class mail in an envelope addressed to: Mail Stop Appeal Brief-Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Typed or printed name	Marilyn Bass		
Signature	Marilyn Bass	Date	November 8, 2004

FEE TRANSMITTAL for FY 2004

Effective 01/01/2004. Patent fees are subject to annual revision.

□ Applicant claims small entity status. See 37 CFR 1.27.

TOTAL AMOUNT OF PAYMENT	($)	340.00

Complete if Known	
Application Number	09/122,349
Filing Date	July 24, 1998
First Named Inventor	Lance Hacking
Examiner Name	Tran, D.
Art Unit	2186
Attorney Docket No.	42390P5965

METHOD OF PAYMENT *(check all that apply)*

☒ Check □ Credit card □ Money Order □ Other □ None

□ Deposit Account

Deposit Account Number: 02-2666

Deposit Account Name: Blakely, Sokoloff, Taylor & Zafman LLP

The Commissioner is authorized to: *(check all that apply)*

☒ Charge fee(s) indicated below □ Credit any overpayments

☒ Charge any additional fee(s) or underpayment of fees as required under 37 CFR §§ 1.16, 1.17, 1.18 and 1.20.

□ Charge fee(s) indicated below, **except for the filing fee** to the above-identified deposit account

FEE CALCULATION

1. BASIC FILING FEE

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
1001	790	2001	395	Utility filing fee	
1002	350	2002	175	Design filing fee	
1003	550	2003	275	Plant filing fee	
1004	790	2004	395	Reissue filing fee	
1005	160	2005	80	Provisional filing fee	

SUBTOTAL (1) ($)

2. EXTRA CLAIM FEES

			Extra Claims		Fee from below		Fee Paid
Total Claims		- 38** =		X		=	
Independent Claims		- 9 =		X		=	
Multiple Dependent						=	

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description
1202	18	2202	9	Claims in excess of 20
1201	88	2201	44	Independent claims in excess of 3
1203	300	2203	150	Multiple Dependent claim, if not paid
1204	88	2204	44	**Reissue independent claims over original patent
1205	18	2205	9	**Reissue claims in excess of 20 and over original patent

SUBTOTAL (2) ($)

***or number previously paid, if greater, For Reissues, see below*

FEE CALCULATION (continued)

3. ADDITIONAL FEES

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
1051	130	2051	65	Surcharge - late filing fee or oath	
1052	50	2052	25	Surcharge - late provisional filing fee or cover sheet.	
2053	130	2053	130	Non-English specification	
1812	2,520	1812	2,520	For filing a request for *ex parte* reexamination	
1804	920 *	1804	920 *	Requesting publication of SIR prior to Examiner action	
1805	1,840 *	1805	1,840 *	Requesting publication of SIR after Examiner action	
1251	110	2251	55	Extension for reply within first month	
1252	430	2252	215	Extension for reply within second month	
1253	980	2253	490	Extension for reply within third month	
1254	1,530	2254	765	Extension for reply within fourth month	
1255	2,080	2255	1,040	Extension for reply within fifth month	
1404	340	2401	170	Notice of Appeal	
1402	340	2402	170	Filing a brief in support of an appeal	340.00
1403	300	2403	150	Request for oral hearing	
1451	1,510	2451	1,510	Petition to institute a public use proceeding	
1452	110	2452	55	Petition to revive - unavoidable	
1453	1,370	2453	685	Petition to revive - unintentional	
1501	1,370	2501	685	Utility issue fee (or reissue)	
1502	490	2502	245	Design issue fee	
1503	660	2503	330	Plant issue fee	
1460	130	2460	130	Petitions to the Commissioner	
1807	50	1807	50	Processing fee under 37 CFR 1.17(q)	
1806	180	1806	180	Submission of Information Disclosure Stmt	
8021	40	8021	40	Recording each patent assignment per property (times number of properties)	
1809	790	1809	395	Filing a submission after final rejection (37 CFR § 1.129(a))	
1810	790	2810	395	For each additional invention to be examined (37 CFR § 1.129(b))	
1801	790	2801	395	Request for Continued Examination (RCE)	
1802	900	1802	900	Request for expedited examination of a design application	

Other fee (specify) ____________________

* Reduced by Basic Filing Fee Paid

SUBTOTAL (3) ($) 340.00

SUBMITTED BY

Complete (if applicable)

Name *(Print/Type)*	Joseph Lutz	*Registration No. (Attorney/Agent)*	43,765	*Telephone*	(310) 207-3800
Signature				*Date*	11/08/04

Based on PTO/SB/17 (10-03) as modified by Blakely, Sokoloff, Taylor & Zafman (wlr) 02/10/2004.
SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450

Attorney Ref.: 042390.P5965

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application for:

Lance Hacking et al.

Application No.: 09/122,349

Filed: July 24, 1998

For: **METHOD AND APPARATUS FOR PERFORMING CACHE SEGMENT FLUSH AND CACHE SEGMENT INVALIDATION OPERATIONS**

Examiner: Denise Tran

Art Group: 2186

APPEAL BRIEF

Mail Stop Appeal Brief - Patent
Commissioner for Patents
P. O. 1450
Alexandria, VA 22313-1450

Dear Sir:

Applicants submit the following Appeal Brief pursuant to 37 C.F.R. §41.37(c) for consideration by the Board of Patent Appeals and Interferences. Applicants also submit herewith a check in the amount of $340.00 to cover the cost of filing the opening brief as required by 37 C.F.R. § 1.17(f). Please charge any additional amount due or credit any overpayment to deposit Account No. 02-2666.

TABLE OF CONTENTS

Page

I. REAL PARTY IN INTEREST

Lance Hacking, Shreekant S. Thakkar, Thomas Huff, Vladimir Pentkovski and Hsien-Cheng E. Hsieh, the parties named in the caption, transferred their rights to that which is disclosed in the subject application through an assignment recorded on July 24, 1998 (9351/0899) in the patent application to Intel Corporation, of Santa Clara, California. Thus, as the owner at the time the brief is being filed, Intel Corporation, of Santa Clara, California is the real party in interest.

II. RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences which will affect or be affected by the outcome of this appeal.

III. STATUS OF CLAIMS

Claims 1, 2, 4-12 and 38-64 are pending and rejected in this application. Applicants hereby appeal the rejection of all pending claims.

IV. STATUS OF AMENDMENTS

The claims are amended in accordance with the Response Amendment filed on February 20, 2004, wherein Claims 1 and 7 were amended. The claim amendments requested in the Response Amendment filed on February 20, 2004 regarding Claims 1 and 7 were entered.

V. SUMMARY OF THE CLAIMED SUBJECT MATTER

The pending claims relate to a method and apparatus for including instructions for performing cache memory invalidate and cache memory flush operations in a computer system. Independent system Claims 1 and 38 recite an execution unit 142 that operates on data elements in a data operand containing a portion of a user specified starting address to invalidate or flush data (as recited by independent system Claim 7) in a predetermined portion of a plurality of cache lines beginning at the user specified starting address in response to receiving a single instruction 160 of a processor instruction set 155, as shown in FIG. 1. (*See*, Applicants' Specification, pg. 4.)

In the embodiment illustrated in FIG. 1, instruction set 155 includes cache control instructions 160. In one embodiment, the cache control instructions include a cache segment invalidate instruction, a cache segment flush instruction and a cache segment flush and invalidate instruction. Applicants' specification provides an example of a cache segment invalidate instruction in the form of a page invalidate (PGINVD) instruction, which operates on a user specified linear address and invalidates the 4 Kbyte physical page corresponding to the linear address from all levels of the cache hierarchy for all agents in the computer system that are connected to the computer system, as recited by dependent Claims 6, 11, 41, 45, 49, 55 and 60. (*See*, Applicants' Specification, pp. 8-9.)

Independent Claim 42 recites a processor 105 including a decoder 140 to decode instructions and a circuit coupled to the decoder 140. As shown in FIGS. 1, 4A and 5A, circuit 140, in response to a single decoded instruction 160 of a processor instruction set 155 reads a portion of an address located in a register specified in the decoded instruction 160 to obtain a user specified starting address of a predetermined area (Z) (*See*, FIG. 4A) of a cache memory (144, 145) on which the instruction will be performed. In response to the instruction, the circuit 140 invalidates in the predetermined area Z of cache memory (144, 145).

Independent Claim 46 recites a processor 105 including a decoder to decode instructions and a circuit 142 coupled to the decoder 140. As shown in FIGS. 1, 4B and 5B, the circuit 142, in response to a single decoded instruction 160 of a processor instruction set 155 is to read a portion of an address 162 located in a register specified in the decoded instruction 160 to obtain a user specified starting address. As shown in FIG. 4B, the user specified starting address identifies a predetermined area of cache memory to flush to, for example, storage device 110. Accordingly, in response to the single decoded instruction from a processor instruction set, circuit 142 copies data into predetermined area of cache memory and stores the copied data in a storage area 110 separate from the cache memory.

Method Claims 51 and 62 are illustrated with reference to FIGS. 3A, 4A and the flowchart shown in FIG. 5A to perform invalidation of the predetermined portion of cache memory. Method Claims 56 and 63 recite features illustrated with reference to FIGS. 3, 4B and the flowchart shown in FIG. 5B to flush the predetermined portion of the cache memory. FIGS. 2 and 3 of Applicants' specification illustrates the general operation of cache control instruction 160 according to one embodiment. In one embodiment, the cache control instruction 160 operates on one page of data stored in the cache of which the beginning address 312 (FIG. 3) is stored in a register 160 (or memory) location specified in the operand 212 (FIG. 2) of the cache control instruction 160. (*See*, Applicants' Specification, pg. 10, line 17 - pg. 11, line 7.)

In other words, as described in Applicants' specification, the portion of memory which is invalidated and/or flushed from cache memory is predetermined and may be, for example, a page or other desired portion of cache memory. Likewise, the cache segment for flushing data is, for example, a page that can be a predetermined number of cache lines, beginning at the user specified starting address. However, it should be recognized this address is predetermined and therefore does not depend on a memory range as specified, for example, by a starting and ending address. In other words, the specification of the user specified starting address is all that is required of the cache control instruction since the instruction operates on a predetermined portion, such as a page.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

The grounds of rejection involved in this appeal are as follows:

Are Claims 1-2, 4-12 and 38-64 unpatentable under 35 U.S.C. §102(e) as obvious over Rahman, U.S. Patent No. 5,778,431 ("Rahman") in view of Milburn, U.S. Patent No. 5,524,433 ("Milburn")?

VII. ARGUMENT

A. Overview of the Cited References

1. Overview of Rahman Reference

Rahman describes a computer system for selectively invalidating the cache lines of cache memory in response to the removal, modification or disabling of system resources, such as, for example, an external memory. (*See* Abstract.) Based on the background description provided in Rahman, the selective flushing technique of cache memory is provided to eliminate the complete flush of a cache memory any time an external memory device is removed from the system. (*See* col. 2, lines 50-53.)

Rahman describes three techniques for performing selective flushing of cache memory in response to the detection of removal of an external memory card. Within each of the techniques, the external memory card, based on its configuration, provides a certain memory capacity. Based on the memory capacity, the external memory is assigned an address range. That address range is compared to tag address values of the cache memory to determine whether a cache line of the cache memory contains data of the external memory in response to detected removal of the external cache memory. When such is detected, each cache line containing data from the external memory is flushed from the cache memory. (*See* col. 3, lines 10-25.)

As an alternative to the hardware implementation, Rahman describes a microcode implementation to perform a similar process

> As an alternative to his hardware implementation of the present invention, instructions may be written in microcode to perform a similar process. Thus, the start and end address values of the external memory device would be fetched and compared through software routines with the tag address values. An instruction to flush a particular line in the cache memory then would be generated by the process in response to an affirmative comparison. (*See* col. 3, lines 26-33.) (Emphasis added.)

In a further alternative embodiment, a bus interface unit is provided containing an address map of available addresses in the external memory device, which are compared to tag addresses in the cache memory to identify matching tag addresses. Accordingly, each cache line corresponding to matching tagged address is invalidated. (*See* col. 3, lines 26–45.) The teachings of Rahman are limited to performing of the cache line invalidation and flushing in response to the removal, modification or disabling of an external memory device.

In addition, invalidating within Rahman is not based on a predetermined portion. The invalidated portion in Rahman will vary based on the physical start and end addresses provided by the external memory and the contents of the cache once the external memory is removed. In Rahman, invalidating begins as dictated by the start address of the external memory and is clearly not a user specified parameter.

2. Overview of Milburn Reference

Milburn relates to a cache control method and mechanism for an external cache memory having multiple cache lines using intra-agent communications to cause invalidating the external cache memory, flushing the external cache memory and/or changing the cache coherency state of lines in the external cache memory. (*See*, Applicants' Abstract.)

In other words, as described within the Background of Milburn:

> It would be desirable to take advantage of the faster local bus to write back the data in the cache. It would also be desirable to flush the cache without relying on the use of external mechanisms, such as a cache controller. Also, it is desirable to flush more than one line of the cache at a time. (col. 2, lines 44-49.)

As further described by Milburn:

> The present invention provides a method and mechanism for controlling an external cache. The present invention controls the external cache using an IAC. (col. 9, lines 26-28.)

As further described within Milburn:

> An inner agent communication (IAC) is a special message sent by the processor of the present invention to itself or to another processor. In the present invention, IACs are used for processor management. In the present invention, one of the utilizations of an IAC message is to control the external cache . . . The currently preferred embodiment of the present invention includes a flush/modify coherency of external cache IAC for controlling the cache. A memory manager in the processor uses the cache control IAC to manage its external cache. (col. 8, lines 42-58.)

However, in contrast to an instruction in a processor instruction set, as described by Milburn:

> In the present invention, the message sent functions like an opcode, therein identifying the operation to be performed. (col. 9, lines 3-4.)

Milburn further describes various fields in the IAC message, for example:

> Field 3 and field 4 provide the address range upon which the operation specified in the control bits are to be performed. By being able to specify an address range instead of a single cache line address, the present invention can flush, modify the coherency state or invalidate portions or all of an external cache using one command. (col. 10, lines 10-15.)

Hence, the use of the address range, based on field 3 and field 4, as specified in an IAC message, will yield a varying range upon which to flush or invalidate within the external cache. Likewise, the inner agent communication, or IAC, message is sent by the processor and hence not by a user.

B. <u>Rejection of Claims 1, 2, 4, 5, 38-40, 42-44, 51-54 and 62 as Obvious Over Rahman in View of Milburn</u>

The Examiner rejected all pending claims, including Claims 1, 2, 4, 5, 38-40, 42-44, 51-54 and 62 under 35 U.S.C. §103(a) as being unpatentable over <u>Rahman</u> in view of <u>Milburn</u>.

1. <u>Errors of Law and Fact in the Rejection</u>

For the reasons provided below, the Examiner has failed to establish a *prima facie* case of obviousness in view of the references of record. The Federal Circuit Court of Appeals in <u>In re Rijckaert</u>, 9 F.3d 1531, 28 U.S.P.Q. 2d 1955 (Fed. Cir. 1993) held that:

> In rejecting claims under 35 U.S.C. § 103, the examiner bears the initial burden of presenting a *prima facie* case of obviousness. . . . "A *prima facie* case of obviousness is established when the teaching from the prior art itself would appear to have suggested the claimed subject matter to a person of ordinary skill in the art.". . . If the examiner <u>fails to establish a *prima facie* case</u>, the <u>rejection</u> is <u>improper</u> and will be overturned. (Emphasis added.) 9 F.3d at 1532, 28 U.S.P.Q. 2d at 1956.

Hence, the case law is quite clear in establishing that the combination of references cited by an Examiner must teach each and every feature of the claimed invention. <u>Id</u>. The Examiner recognizes <u>Rahman</u>'s failure to teach a single instruction of a processor instruction set to direct the selective cache line invalidation taught by <u>Rahman</u>. As a result, the Examiner cites <u>Milburn</u>. According to the Examiner:

> <u>Milburn</u> shows providing a single instruction with an operation code and address values in its operand (e.g., col. 9, lines 1-10; col. 10, lines 10-15; and col. 11, lines 1-20) are well known and expected in the art. It would have been obvious to one of ordinary skill in the art at the time the invention was made to provide a single instruction with an operation code and address values in its operand to <u>Rahman</u>'s system because it would allow an instruction to influence multiple lines in a cache, invalidating an entire cache range, thereby reducing overall operation times. (Final Office Action, mailed May 5, 2004, pg. 4.)

According to the Examiner, the concept and advantages of providing a single instruction with an operation code and address values in its operand are well known and expected in the art. However, Milburn does not teach a single instruction of a processor instruction set. As described within Milburn:

> An inner agent communication (IAC) is a special message sent by the processor of the present invention to itself or to another processor. In the present invention, IACs are used for processor management. In the present invention, one of the utilizations of an IAC message is to control the external cache. (col. 8, lines 42-47.) (Emphasis added.)

Applicants respectfully submit that one skilled in the art would not recognize the IAC message of Milburn, which is used for processor management, as an instruction from a processor instruction set. As taught by Milburn, such techniques for performing processor management, are available to the system architect but are certainly not added to the processor instruction set, which may be used by a user or programmer of the computer system.

Furthermore, the techniques described by Rahman are performed by the computer system in response to the computer system detection of removal, modification or disabling of system resources. Hence, the techniques described by Rahman and Milburn are performed without the knowledge of the user and therefore do not need to be made available to the user in the form of a single instruction from a processor instruction set. Accordingly, Applicants submit that one skilled in the art would not perform the selective invalidating of the cache memory of Rahman in view of Milburn by providing a single instruction within a processor instruction set to a user, as recited by the claimed invention.

In fact, one cannot find obviousness through hindsight to construct a claimed invention from elements of the prior art. In re Warner, 379 F.2d 1011, 1016, 154 USPQ 173, 177 (C.C.P.A. 1967.) Applicants respectfully submit that the features of the claimed invention could only be arrived at through inappropriate hindsight. Hence, a *prima facie* case of obviousness has not been established and the rejection is erroneous and should be overturned.

2. Specific Limitations Not Described in the Prior Art

Independent Claims 1, 38, 42, 51 and 62 recite analogous claim features. Claim 1 is representative. Independent Claim 1 recites the following claim feature, which is neither taught nor suggested by either Rahman, Milburn or the references of record:

> an execution unit coupled to said storage area to operate on data elements in said data operand containing a portion of a user specified starting address to invalidate data in a predetermined portion of the plurality of cache lines beginning at the user specified starting address in response to receiving a single instruction of a processor instruction set. (Emphasis added.)

3. Explanation Why Such Limitations Render the Claims Non-obvious Over the Prior Art

According to the Examiner, the user specified start address at which to begin invalidation, as recited by the claimed invention, is taught by the combination of Rahman in view of Milburn. Applicants respectfully disagree with the Examiner's contention based on careful review of Rahman as well as Milburn.

According to the Examiner:

> Rahman shows user-specified starting address (e.g., col. 3, lines 25-35 or Abstract). In other words, based on the citation above, Rahman teaches the software routines or instructions in microcode written by a user who specified the start address to compare to the tag address. (*See* Final Office Action mailed May 5, 2004, pg. 8.)

Applicants respectfully disagree with the Examiner's contention. Applicants respectfully submit that the passage cited above, which is further described at col. 7, lines 22-27, describe details for implementing the selective flushing of cache memory, as taught by Rahman. Such implementation details are performed by the system architect and are certainly not accessible to a user via, for example, one or more instructions of the processor instruction set. Likewise, the IAC message taught by Milburn to provide processor management, such as controlling the external cache, is performed at the direction of the processor; and hence, does not involve a user, user activity or even user knowledge thereof to enable user specification of the start address as recited by the claimed invention.

In addition, as indicated by the cited passage:

> The start and end address values of the external memory device would be fetched and compared through software routines with the tag address values. (col. 3, lines 31-33.)

Applicants respectfully submit that fetching of start and end address values of the external memory device does not teach or suggest either a user-specified or a user-definable starting address, as recited by the claimed invention. The start address, as taught by Rahman, is based on the physical address range to be provided by the external memory device as assigned by, for example, an operating system (OS). A user of the system taught by Rahman cannot specify the start address that is assigned to the external memory device by the OS. Therefore, the user does not specify the start address, as contended by the Examiner.

Consequently, Applicants submit that the teachings of Rahman in view of Milburn fail to teach or suggest the user specified starting address at which to begin invalidation, as recited by the claimed invention.

Furthermore, the claimed invention recites invalidating of data in a predetermined cache memory portion. By way of contrast, both Rahman and Milburn teach the selective invalidation of the cache lines of a cache memory. Within Rahman, each tag address of the cache memory is compared to the physical address range of the external memory in response to removal of the external memory. Similarly, in Milburn, the address range provided with the IAC message is compared to each tag address of cache memory to selectively identify cache lines having a match tag address.

As known to those skilled in the art, cache memory is used to store the most recently used data. In addition, such data is stored in the first available cache line and hence, is not placed in any specific order. Accordingly, even given an address range, such as provided by the IAC message, as taught by Milburn, or the physical address range of an external memory device, as taught by Rahman, the non-contiguous nature of cache memory virtually guarantees that a predetermined portion of the cache memory is not invalidated. In other words, the teachings of both Rahman and Milburn require a comparison of each tag address to the provided address range; for those matching tag addresses, the corresponding cache line is evicted.

Therefore, Applicants respectfully submit that the combination of Rahman in view of Milburn fails to teach the invalidation of a predetermined cache memory portion since cache lines corresponding to the address ranges provided by either Milburn or Rahman may have been evicted from the cache to make room for more recently used data or possibly were never cached, and therefore would not result in the invalidation of a predetermined portion, as recited by the claimed invention.

Moreover, Claim 1 recites that the invalidating is performed in response to receiving a single instruction of a processor instruction set. Conversely Rahman performs its invalidation and flushing in response to detected removal of the external memory. (*See*, col. 6, lines 26-30.) Likewise, Milburn teaches an IAC message for processor management, such as controlling an external cache. Applicants submit that the selective invalidating of cache lines of an external cache memory based on a memory range of an external memory or a memory range provided by an IAC message, as taught by Rahman and Milburn, is not performed in response to a single instruction of a processor instruction set, as recited by the claimed invention.

Consequently, even if Milburn disclosed cache invalidation according to a single instruction of a processor instruction set, the Examiner fails to provide a motivation to combine the missing elements provided by Milburn within the teachings of Rahman since both references are directed to computer system activity that is unknown to the user. Accordingly, Applicants respectfully submit that the features of the claimed invention could only be arrived at through inappropriate hindsight.

Accordingly, Applicants respectfully submit that the Examiner has failed to establish a *prima facie* case of obviousness of the claimed invention since the combination of references of Rahman in view of Milburn fail to teach or suggest each of the features of the claimed invention, including, but not limited to, specification of a starting address by a user, invalidation of data in a predetermined portion of a plurality of cache lines beginning at the starting address and such invalidation performed in response to a single instruction of a processor instruction set. Accordingly, Applicants respectfully submit that the teachings from the combination of Rahman in view of Milburn would not have suggested the claimed subject matter to a person of ordinary skill in the art. As a result, the Examiner has failed to establish a *prima facie* case of obviousness and therefore the rejection is improper and should be overturned. Id.

Therefore, Applicants respectfully submit that a *prima facie* case of obviousness is not established and therefore the rejection of Claims 1, 2, 4, 5, 38-40, 42-44, 51-54 and 62 is erroneous and should be overturned. Accordingly, Applicants respectfully request that the §103(a) rejection of Claims 1, 2, 4, 5, 38-40, 42-44, 51-54 and 62 be overturned.

C. Rejection of Claims 7-10, 12, 46-48, 50, 56-59, 61, 63 and 64 as Obvious over Rahman in View of Milburn

1. Errors of Law and Fact in the Rejection

The Examiner has made the same errors as described previously with respect to rejected independent Claims 1, 38, 42, 51 and 62. In addition, the Examiner has failed to show that the prior art references of Rahman and Milburn teach or suggest all claim features of Claim 7.

According to the Examiner:

> It would have been obvious to one of ordinary skill in the art at the time the invention was made to provide a single instruction with an operation code and address values in its operand to Rahman's system because it would allow an instruction to influence multiple lines in a cache, invalidating an entire cache range, thereby reducing overall operations times. (*See* Office Action, pg. 4.) (Emphasis added.)

However, the techniques described by Rahman are performed by the computer system in response to the computer system detection of removal, modification or disabling of system resources. Likewise, the IAC message taught by Milburn to provide processor management, such as controlling the external cache, is performed at the direction of the processor; and hence, does not involve a user, user activity or even user knowledge thereof. As a result, Applicants submit that one skilled in the art would not perform the selective invalidating, as taught by Rahman in view of Milburn, using a single instruction of the processor instruction set, since the techniques or instructions to perform such activity are solely performed at the direction of a computer system.

In other words, the techniques described by Rahman and Milburn are performed without the knowledge of the user and therefore do not need to be made available to the user in the form of a single instruction from a processor instruction set. Accordingly, Applicants submit that one skilled in the art would not perform the selective invalidating of the cache memory of Rahman in view of Milburn by providing a single instruction within a processor instruction set to a user, as recited by the claimed invention due to the absence of any motivation for the modification.

Furthermore, as indicated above, the teachings of Rahman and Milburn are limited to selective invalidating cache lines of cache memory for which a tag address matches either a physical address provided by an external memory or an address range provided in an IAC message. Applicants respectfully submit that the selective flushing of cache memory, as taught by Rahman and Milburn, teaches away from the invalidation of a predetermined portion of cache memory, as recited by the claimed invention, since invalidation of the predetermined portion could lead to invalidation and flushing of data from system memory, which is stored within cache 106, as taught by Rahman. Hence, modification of the selective invalidation, as taught by Rahman, could render Rahman unsatisfactory for its intended purpose by causing the flushing of data stored within cache memory, which is from system memory 114.

Therefore, Applicants respectfully submit that the modification of Rahman in view of Milburn, as proposed by the Examiner, to render the claimed invention obvious would run contrary to the explicit teachings of Rahman. One of ordinary skill in the art would not be motivated to modify Rahman in a manner specifically contrary to Rahman's own teaching. Consequently, the teachings of Rahman in view of Milburn would not have suggested the claimed subject matter to a person of ordinary skill in the art, as required to establish a *prima facie* case of obviousness. Id. Accordingly, as required by case law, the Examiner has failed to establish a *prima facie* rejection and therefore, the rejection of the claimed invention is improper and should be overturned. Id.

Hence, a *prima facie* case of obviousness has not been established and the rejection is erroneous and should be overturned.

2. Specific Limitations Not Described in the Prior Art

Independent Claims 46, 46 and 63 recite analogous claim features. Claim 7 is representative. Claim 7 recites the following claim feature, which is neither taught nor suggested by either Rahman, Milburn or the references of record:

> an execution unit coupled to said first storage area, said second storage area, and said cache memory, said execution unit to operate on the portion of a user specified address in said data operand to copy data from a predetermined portion of the plurality of cache lines beginning at the user specified starting address in the cache memory to the first storage area, in response to receiving a single instruction of a processor instruction set. (Emphasis added.)

3. Explanation Why Such Limitations Render the Claims Non-obvious Over the Prior Art

As indicated above, the teachings of Rahman and Milburn are limited to selectively invalidating cache lines according to a range of a start and end physical addresses assigned by the OS to a system resource or a varying range provided by an IAC message. Applicants respectfully submit that Rahman and Milburn are devoid of any teaching regarding copying data from a predetermined portion of the plurality of cache lines beginning at the user specified starting address, as recited by the claimed invention.

Accordingly, Applicants respectfully submit that the Examiner has failed to establish a *prima facie* case of obviousness of the claimed invention since the combination of references of Rahman in view of Milburn fail to teach or suggest each of the features of the claimed invention, including, but not limited to, specification of a starting address by a user, flushing of data in a predetermined portion of a plurality of cache lines beginning at the starting address and such flushing performed in response to a single instruction of a processor instruction set. Accordingly, Applicants respectfully submit that the teachings from the combination of Rahman in view of Milburn would not have suggested the claimed subject matter to a person of ordinary skill in the art. As a result, the Examiner has failed to establish a *prima facie* case of obviousness and therefore the rejection is improper and should be overturned. Id.

Therefore, Applicants respectfully submit that a *prima facie* case of obviousness of Claims 7-10, 12, 46-48, 50, 56-59, 61, 63 and 64 is not established and therefore the rejection of Claims 7-10, 12, 46-48, 50, 56-59, 61, 63 and 64 is erroneous and should be overturned. Accordingly, Applicants respectfully request that the §103(a) rejection of Claims 7-10, 12, 46-48, 50, 56-59, 61, 63 and 64 be overturned.

D. Rejection of Claims 6-11, 41, 45, 49, 55 and 60 as Obvious over Rahman in View of Milburn

The Examiner rejected all pending claims, including Claims 6, 11, 41, 45, 49, 55 and 60 under 35 U.S.C. §103(a) as being unpatentable over Rahman in view of Milburn.

1. Errors of Law and Fact in the Rejection

The Examiner has made the same errors as described previously with respect to rejected independent Claims 1, 7, 38, 42, 46, 51 and 56 from which Claims 6, 11, 41, 45, 49, 55 and 60 depend, respectively. In addition, the Examiner has failed to show that the prior art references of Rahman and Milburn teach or suggest all claim features of Claims 6, 11, 41, 45, 49, 55 and 60.

Rahman describes a computer system for selectively invalidating the cache lines of cache memory in response to the removal, modification or disabling of system resources, such as, for example, an external memory. (*See* Abstract.) Based on the background description provided in Rahman, the selective flushing technique of cache memory is provided to eliminate the complete flush of a cache memory any time an external memory device is removed from the system. (*See* col. 2, lines 50-53.)

Rahman describes three techniques for performing selective flushing of cache memory in response to the detection of removal of an external memory card. Within each of the techniques, the external memory card, based on its memory capacity, provides a certain range of physical addresses. As described in FIG. 1 of Rahman:

> . . . each line of cache memory 106 has associated therewith address tag and state information. The address tag indicates a physical address in system memory 114 or in external memory (such as may be present for example in the removable card driver 144) corresponding to each entry within cache memory 106. (col. 5, lines 28-34.) (Emphasis added.)

In response to removal of the external memory, the physical address range provided by the external memory is compared to tag address values of the cache memory to determine whether a cache line of the cache memory contains data of the external memory. When such is detected, each cache line containing data from the external memory is flushed from the cache memory. (*See* col. 3, lines 10-25.)

Due to the sharing of cache memory 106 between system memory 114 and external memory within card driver 144, Rahman teaches the selective flushing of cache lines of cache memory 106 to avoid flushing of cache lines containing data from system memory 114.

Milburn further describes various fields in the IAC message, for example:

> Field 3 and field 4 provide the address range upon which the operation specified in the control bits are to be performed. By being able to specify an address range instead of a single cache line address, the present invention can flush, modify the coherency state or invalidate portions or all of an external cache using one command. (col. 10, lines 10-15.) (Emphasis added.)

Based on the cited passage above, the address range specified within field 3 and field 4 within the IAC message, as taught by Milburn, result in the same selective invalidation as taught by Rahman. In other words, the teachings of Milburn enable specification of an address range instead of a single cache line address, to cause the selective invalidation and/or flushing of each line having a tag address matching the specified range.

Applicants respectfully submit that the selective flushing of cache memory, as taught by Rahman and Milburn, teaches away from the invalidation of a predetermined portion of cache memory, as recited by the claimed invention, since invalidation of the predetermined portion could lead to invalidation and flushing of data from system memory, which is stored within cache 106, as taught by Rahman. Hence, modification of the selective invalidation, as taught by Rahman, could render Rahman unsatisfactory for its intended purpose by causing the flushing of data stored within cache memory, which is from system memory 114.

Applicants respectfully submit that the modification of Rahman in view of Milburn to render the claimed invention obvious would run contrary to the explicit teachings of Rahman. One of ordinary skill in the art would not be motivated to modify Rahman in a manner specifically contrary to Rahman's own teaching. Consequently, the teachings of Rahman in view of Milburn would not have suggested the claimed subject matter to a person of ordinary skill in the art, as required to establish a *prima facie* case of obviousness. Id. Accordingly, as required by case law, the Examiner has failed to establish a *prima facie* rejection and therefore, the rejection of the claimed invention is improper and should be overturned. Id.

Hence, Applicants respectfully submit that the features of Claims 6, 11, 41, 45, 49, 55 and 60 could only be arrived at through inappropriate hindsight. Accordingly, a *prima facie* case of obviousness has not been established and the rejection is erroneous and should be overturned.

2. Specific Limitations Not Described in the Prior Art

Claims 6, 11, 41, 45, 49, 55 and 60 include analogous claim features. Claim 6 is representative. Claim 6 recites the following claim feature, which is neither taught nor suggested by either Rahman, Milburn or the references of record:

> wherein the predetermined portion of the plurality of cache lines is a page in the cache memory.

3. Explanation Why Such Limitations Render the Claims Non-obvious over the Prior Art

Rahman teaches away from the invalidation of a page of cache memory due to the fact that the cache memory taught by Rahman is shared by an external memory device and system memory. As a result, when the external memory device is removed, an address tag of each cache line within cache memory is compared to the physical address range provided by the external memory. When a hit is detected, either an instruction or a signal is provided, as taught by Rahman, to invalidate and flush the matching cache line within cache memory to avoid flush of data cached from the system memory.

Hence, Rahman teaches away from invalidating either a predetermined portion or a page of the lines of cache memory, since such action could cause invalidation and flushing of cached data from system memory, which is also temporarily stored within the cache memory taught by Rahman.

Similarly, the address range specification in the IAC message, as taught by Milburn, would still require a comparison of each tag addresses of the cache memory to the address range specified by the IAC message. Unless a hit is detected in response to the address range provided by the IAC message, the respective cache line would not necessarily be flushed. As known to those skilled in the art, a cache memory stores most recently used data. Here, the most recently used data within the cache memory may be from both system memory and external memory.

Likewise, since cache memory stores data based on the currently available cache lines, cache data from memory is not required to be consecutive within the cache memory. Hence, due to the non-contiguous nature of cache memory, the address range specification, as taught by Rahman in view of Milburn, would not result in the invalidation or flushing of a page within the cache memory. Consequently, invalidating within both Rahman and Milburn is not based on either a predetermined portion or a page of cache memory.

Moreover, Applicants submit that the modification of Rahman in view of Milburn, as proposed by the Examiner, to render the claimed invention obvious would run contrary to the explicit teachings of Rahman. One of ordinary skill in the art would not be motivated to modify Rahman in a manner specifically contrary to Rahman's own teachings. Consequently, since one skilled in the art would not be motivated to modify Rahman in the manner explicitly contrary to Rahman's own explicit teachings, Applicants respectfully submit that the Examiner is prohibited from establishing that the combination of Rahman in view of Milburn would have suggested the claimed subject matter to a person of ordinary skill in the art, as required to establish a *prima facie* case of obviousness. Id.

Accordingly, the Examiner's failure to establish a *prima facie* case of obviousness renders the rejection of the claimed invention improper and requires that the Examiner's rejection of the claimed invention be overturned. Id. Therefore, Applicants respectfully submit that a *prima facie* case of obviousness of Claims 6, 11, 41, 45, 49, 55 and 60 is not established and therefore the rejection of Claims 6, 11, 41, 45, 49, 55 and 60 is erroneous and should be overturned. Accordingly, Applicants respectfully request that the §103(a) rejection of Claims 6, 11, 41, 45, 49, 55 and 60 be overturned.

VIII. CONCLUSION AND RELIEF

Based on the foregoing, Applicant requests that the Board overturn the rejection of all pending claims and hold that all of the claims of the present application are allowable.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN

Dated: November 8, 2004

By:____________________
Joseph Lutz, Reg. No. 43,765

12400 Wilshire Boulevard
Seventh Floor
Los Angeles, California 90025
(310) 207-3800

CERTIFICATE OF MAILING:
I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail, with sufficient postage, in an envelope addressed to: Mail Stop Appeal Brief - Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on November 8, 2004.

____
Marilyn Bass November 8, 2004

IX. APPENDIX

The claims involved in this Appeal are as follows:

1. (Previously Presented) A computer system comprising:

a cache memory having a plurality of cache lines each of which stores data;

a storage area to store a data operand; and

an execution unit coupled to said storage area to operate on data elements in said data operand containing a portion of a user specified starting address to invalidate data in a predetermined portion of the plurality of cache lines beginning at the user specified starting address in response to receiving a single instruction of a processor instruction set.

2. (Original) The computer system of Claim 1, wherein the data operand is a register location.

3. (Cancelled).

4. (Previously Presented) The computer system of Claim 1, wherein the portion of the starting address includes a plurality of most significant bits of the starting address.

5. (Original) The computer system of Claim 4, wherein execution unit shifts the data elements by a predetermined number of bit positions to obtain the starting address of the cache line in which data is to be invalidated.

6. (Original) The computer system of Claim 1, wherein the predetermined portion of the plurality of cache lines is a page in the cache memory.

7. (Previously Presented) A computer system comprising:

a first storage area to store data;

a cache memory having a plurality of cache lines each of which stores data;

a second storage area to store a data operand containing a portion of an address; and

an execution unit coupled to said first storage area, said second storage area, and said cache memory, said execution unit to operate on the portion of a user specified address in said data operand to copy data from a predetermined portion of the plurality of cache lines beginning at the user specified starting address in the cache memory to the first storage area, in response to receiving a single instruction of a processor instruction set.

8. (Original) The computer system of claim 7, wherein the data operand is a register location.

9. (Original) The computer system of claim 8, wherein the register location contains a plurality of most significant bits of a starting address of the cache line in which data is to be copied.

10. (Previously Presented) The computer system of claim 9, wherein execution unit shifts the portion of an address by a predetermined number of bit positions to obtain the starting address of the cache line in which data is to be copied.

11. (Original) The computer system of Claim 7, wherein the predetermined portion of the plurality of cache lines is a page in the cache memory.

12. (Original) The computer system of Claim 7, wherein the execution unit further invalidates data in the predetermined portion of the plurality of cache lines in response to receiving the single instruction, upon copying the data to the first storage area.

Claims 13-37 (Cancelled)

38. (Previously Presented) A computer system comprising:

a cache memory having a plurality of cache lines each of which stores data;

a storage area to store a data operand; and

an execution unit coupled to said storage area to operate on data elements in said data operand identifying a user-definable linear or physical address identifying a predetermined portion of the plurality of cache lines to invalidate data in the predetermined portion of the plurality of cache lines in response to receiving a single cache control instruction of a processor instruction set, the single cache control instruction including a reference to the data operand.

39. (Previously Presented) The computer system of Claim 38, wherein the data operand is a register location.

40. (Previously Presented) The computer system of Claim 39, wherein execution unit shifts the data elements by a predetermined number of bit positions to obtain the starting address of the cache line in which data is to be invalidated.

41. (Previously Presented) The computer system of Claim 38, wherein the predetermined portion of the plurality of cache lines is a page in the cache memory.

42. (Previously Presented) A processor comprising:

a decoder configured to decode instructions; and a circuit coupled to said decoder, said circuit in response to a single decoded instruction of a processor instruction set being configured to:

read a portion of an address located in a register specified in the decoded instruction to obtain a user specified starting address of a predetermined area of a cache memory on which the instruction will be performed; and invalidate in the predetermined area of cache memory.

43. (Previously Presented) The processor of Claim 42, wherein the portion of an address includes a plurality of most significant bits of the starting address.

44. (Previously Presented) The processor of Claim 43, wherein the circuit shifts the portion of an address by a predetermined number of bits positions to obtain the starting address of a cache line of the predetermined area of the cache memory in which data is to be invalidated.

45. (Previously Presented) The processor of Claim 42, wherein the predetermined area of the cache memory comprises a plurality of cache lines forming a page in the cache memory.

46. (Previously Presented) A processor comprising:

a decoder to decode instructions, and

a circuit coupled to said decoder, said circuit in response to a single decoded instruction of a processor instruction set being configured to: read a portion of an address located in a register specified in the decoded instruction to obtain a user specified starting address of a predetermined area of a cache memory on which the instruction will be performed;

copy data in the predetermined area of the cache memory; and

store the copied data in storage area separate from the cache memory.

47. (Previously Presented) The processor of Claim 46, wherein the portion of an address includes a plurality of most significant bits of the starting address.

48. (Previously Presented) The processor of Claim 47, wherein the circuit shifts the portion of the address by a predetermined number of bit positions to obtain the starting address of a cache line of the cache memory in which data is to be copied.

49. (Previously Presented) The processor of Claim 47, wherein the predetermined area comprises a plurality of cache lines forming a page in the cache memory.

50. (Previously Presented) The processor of Claim 47, wherein said circuit further invalidates the data in the predetermined portion of the plurality of cache lines in response to receiving the single instruction, upon copying the data to the storage area.

51. (Previously Presented) A computer-implemented method, comprising:

a) decoding a single instruction of a processor instruction set;

b) in response to said decoding of the single instruction, obtaining a portion of a user specified starting address of a predetermined area of a cache memory on which the single instruction will be performed by reading a portion of an address contained in a storage location specified in the decoded instruction; and

c) completing execution of said single instruction by invalidating data in the predetermined area of the cache memory.

52. (Previously Presented) The method of Claim 51, wherein c) comprises setting an invalid bit corresponding to the predetermined area of the cache memory.

53. (Previously Presented) The method of Claim 51, wherein b) comprises:

shifting the portion of the starting address by a predetermined number of bit positions to obtain the starting address of a cache line of the cache memory in which data is to be invalidated.

54. (Previously Presented) The method of Claim 53, wherein the portion of the starting address contains a plurality of most significant bits of the starting address, and the predetermined number of bit positions represent the number of least significant bits of the starting address.

55. (Previously Presented) The method of Claim 51, wherein the predetermined area is a page in the cache memory.

56. (Previously Presented) A computer-implemented method, comprising:

a) decoding a single instruction of a processor instruction set;

b) in response to said decoding the single instruction, obtaining a portion of a user specified starting address of a predetermined area of a cache memory on which the single instruction will be performed by reading a portion of an address contained in a storage location specified in the decoded instruction; and

c) completing execution of said single instruction by copying data in the predetermined area of cache memory and storing the copied data in a storage area separate from the cache memory.

57. (Previously Presented) The method of Claim 56, wherein c) comprises setting an invalid bit corresponding to the predetermined area of the cache memory.

58. (Previously Presented) The method of Claim 56, wherein b) comprises:

shifting the portion of the starting address by a predetermined number of bit positions to obtain the starting address of a cache line associated with the predetermined area.

59. (Previously Presented) The method of Claim 58, wherein the portion of the starting address contains a plurality of most significant bits of the starting address, and the predetermined number of bit positions represent the number of least significant bits of the starting address.

60. (Previously Presented) The method of Claim 56, wherein the predetermined area comprises a plurality of cache lines forming a page in the cache memory.

61. (Previously Presented) The method of Claim 56, further comprises:

d) invalidating the data in the predetermined area in response to receiving the single instruction, upon copying the data to the storage area.

62. (Previously Presented) A computer-readable apparatus, comprising:

a computer-readable medium that stores an instruction which when executed by a processor causes said processor to:

a) decode a single instruction of a processor instruction set;

b) in response to decoding the single instruction, obtain a portion of a user specified starting address of a predetermined area of a cache memory on which the single instruction will be performed by reading a portion of an address contained in a storage location specified in the decoded instruction; and

c) complete execution of said single instruction by invalidating data in the predetermined area of the cache memory.

63. (Previously Presented) A computer-readable apparatus comprising:

a computer-readable medium that stores an instruction which when executed by a processor causes said processor to:

a) decode a single instruction of a processor instruction set;

b) in response to decoding the single instruction, obtain a portion of a user specified starting address of a predetermined area of a cache memory on which the single instruction will be performed by reading a portion of an address contained in a storage location specified 8 in the decoded single instruction; and

c) complete execution of said single instruction by copying data in the predetermined area of the cache memory and storing the copied data in a storage area separate from the cache memory.

64. (Previously Presented) The apparatus of Claim 63, wherein the instruction further causes the processor to:

invalidate the data in a predetermined portion of a plurality of cache lines forming the predetermined area of the cache memory in response to receiving the instruction, upon copying the data to the storage area.